BBN Systems and Technologies
A Division of Bolt Beranek and Newman Inc.

BBN Report No. 7528

Contract Number N00014-85-C-0279
Final Report
18 January 1988 - 30 September 1990

IMPROVEMENTS IN THE BYBLOS CONTINUOUS SPEECH RECOGNTION SYSTEM

R. Schwartz, S. Austin, C. Barry, F. Kubala, J. Makhoul,
P. Placeway, G. Yu

November 1990

BBN Report No. 7528

ARPA Order Number 4707
Contract Number N00014-85-C-0279
Contract Duration: 17 Jan 1985 - 30 September 1990
Principal Investigator: J. Makhoul, (617)873-3332

Final Report

IMPROVEMENTS IN THE BYBLOS CONTINUOUS SPEECH RECOGNTION SYSTEM

R. Schwartz, S. Austin, C. Barry, F. Kubala, J. Makhoul,
P. Placeway, G. Yu

November 1990

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1990

3. REPORT TYPE AND DATES COVERED
Final Report, 1/88 - 9/90

4. TITLE AND SUBTITLE
Improvements in the BYBLOS Continuous
Speech Recognition System

5. FUNDING NUMBERS
N00014-85-C-0279

6. AUTHOR(S)

R. Schwartz, S. Austin, C. Barry, F. Kubala,
J. Makhoul, P. Placeway, G. Yu

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

BBN Systems and Technologies
10 Moulton Street
Cambridge, MA 02138

8. PERFORMING ORGANIZATION REPORT NUMBER
Report No. 7528

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Department of the Navy
Arlington, Virginia 22217-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution of the document is unlimited. It may be released to the Clearinghouse, Dept. of Commerce, for sale to the general public.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of this basic research was to develop accurate mathematical models of speech sounds for the purpose of large-vocabulary continuous speech recognition. The research has focussed on three areas: developing better speech models to improve recognition accuracy, exploring new techniques for speaker-independent training, and developing speaker adaptation techniques that allow system use with a minimum of training. The work was performed within the BBN BYBLOS speech recognition system, which is based on the use of phonetic hidden Markov models. As a result of several model improvements, we have succeeded in decreasing the word error rate by a factor of four for speaker-dependent and speaker-independent recognition. In speaker-independent recognition, we have developed a new training paradigm in which we record speech from only a dozen speakers instead of the traditional approach of recording more than a hundred speakers. The same approach has been shown to be useful for effective speaker adaptation with only two minutes of speech training.

14. SUBJECT TERMS
continuous speech recognition, hidden Markov models, speaker-independent recognition, speaker adaptation

15. NUMBER OF PAGES
54

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT

Contents

Chapter 1

Executive Summary

This is the final report for the project Research in Continuous Speech Recognition, sponsored by the Defense Advanced Research Agency (DARPA) and monitored by ONR under Contract No. N00014-85-C-0279. The report covers the period 18 Jaiuary 1988 to 30 September 1990; an earlier report covered the first three years of the project [33].

The objective of this basic research is to develop accurate and detailed mathematical models of the fundamental units of speech (phonemes) for the purpose of large-vocabulary continuous speech recognition. The important goals of this work are to achieve the highest possible word recognition accuracy in continuous speech and to develop methods for the rapid adaptation of phonetic models to the voice of a new speaker.

The research during the past three years can be categorized into three broad topics: developing better speech models to improve recognition accuracy, exploring new techniques for speaker-independent training, and developing speaker adaptation techniques that allow the system to be used by new speakers with a minimum amount of training. The research performed under each of these areas is summarized below.

Our primary goal in this project has been to increase speech recognition accuracy. The fundamental means by which we can improve accuracy is by improving our model of speech in some way. Therefore, we have explored several modifications of the speech model and techniques for parameter estimation. The result has been that, during the course of this three year effort, we have reduced the word error rate for speech recognition by a factor of 4, as measured on a standard DARPA corpus. For example, when using speaker-dependent models, the word error has decreased from 7.5% in 1987 to 1.7%. In speaker-independent recognition, the error rate has decreased from 16% to 3.9%. The current error rates for speaker-dependent and speaker-independent recognition are the lowest reported thus far for this corpus. Details of how we achieved this large improvement in performance can be found in Chapter 2.

In addition to our improved results on speaker-independent recognition, we have developed a novel paradigm for training a speaker-independent system. Previously it had been thought that, to obtain reasonable speaker-independent performance, it was necessary to collect speech from a large number of speakers. Typically, a few minutes of speech is collected from each of at least 100 speakers. For example, the speaker-independent experiments performed with the DARPA corpus have typically used the speech of 109 speakers to train the phonetic models. Using our new paradigm for speaker-independent training, we have found that it is possible to obtain good speaker-independent performance by training on speech taken from only a few speakers (12 speakers in our experiments), but with 30 minutes of speech taken from each of the speakers. In addition, we developed a new training method in which we estimate models for each of the speakers separately, and then combine the models to obtain the speaker-independent model. There are several important advantages to this new training paradigm. First, it requires much less effort to collect speech from a few speakers. Second, it is possible to add models from new speakers incrementally, without the need to reprocess the speech of the other speakers. And finally, in contrast to the original paradigm, those speakers who have provided a substantial amount of speech will have the benefit of speaker-dependent recognition with performance three times better than with the speaker-independent model. Chapter 3 describes the new training paradigm in more detail.

The third major area of research at BBN has been on techniques for rapid speaker adaptation. There are two important goals for speaker adaptation. The first deals with the need to minimize the amount of speech needed for training. On the one hand, speaker-dependent recognition requires the collection of about 30 minutes from each new speaker. On the other hand, the initial effort required to collect speech from a large number of speakers for a speaker-independent model may be prohibitive for each new application. In contrast, using speaker adaptation, we start with a speaker-dependent model from only a single reference speaker, and then adapt that model to a new speaker by using only 2 minutes of speech from the new speaker. During this effort we have improved the accuracy of our previous methods for speaker adaptation. The accuracy is now equivalent to that obtained with speaker-independent models, but at a much lower training cost.

The second goal of speaker adaptation is to improve the performance over that obtainable with speaker-independent models. The large difference between the accuracy with speaker-dependent and speaker-independent models points to the need for a way to improve recognition accuracy for a given speaker quickly. In this area, we have developed the first successful method for speaker adaptation starting from a speaker-independent speech corpus. The error rate using this technique is reduced by almost a factor of two relative to the speaker-independent error rate. Details of this work are provided in Chapter 4.

Chapter 2

Improved Speech Models

All the work in this project has been performed within the context of the BBN BYBLOS continuous speech recognition system. We extended the BYBLOS system in several ways in an attempt to provide more detailed acoustic-phonetic information: robust smoothing techniques, supervised vector quantization, phonetic HMM topology, modeling coarticulation between words, estimation of codebook weights, and Ear-Model signal processing.

2.1 Robust Smoothing Methods

In this section we present three methods for smoothing discrete probability functions as used in discrete hidden Markov models for large vocabulary continuous speech recognition. The smoothing is based on deriving a probabilistic cooccurrence matrix between the different vector-quantized spectra. Each estimated probability density is then multiplied by this matrix, ensuring that none of the probabilities are severely underestimated due to lack of training data. Experimental results show a 20%-30% reduction in error rate when this smoothing is used.

2.1.1 Introduction

Much of the research in speech recognition is devoted to improving the structure of the statistical model of speech. Frequently, improving the model involves increasing the complexity or dimensionality of the model. For example, we use context-dependent phonetic models, which increases the number of models. We add features, such as spectral derivatives, which increases the dimensionality of the feature space. We use a non-parametric probability density function (pdf) to have flexibility in the model, but we lose the benefit of the compactness of a parametric model. Each of these improvements comes with an increase in the effective number of degrees of freedom in our model. Unfortunately, more training data is needed to estimate reliably the increased number of free parameters. Conversely, faced with a fixed amount of training data, we must limit the number of free parameters or else our "improvements" will not be realized.

In the BYBLOS system, we use discrete nonparametric pdfs of context-dependent phonetic models. Most of these pdfs are trained with only a few tokens of speech (typically between 1 and 10). These discrete distributions work surprisingly well, given the small amount of training. However, they are certainly prone to the problem of spectral types that do not appear in the training set for a given model, but are, in fact, likely to occur for that model. One way to determine the magnitude of this problem is to compare the recognition rate when the system is tested on the training data and on independent test data. If the difference in accuracy is large, then there is not enough training data for the model.

There are many techniques for avoiding the problem of unobserved spectra. The most common is to combine (average) the estimated discrete pdf with an alternate model, perhaps with a weight that depends on the number of training tokens [16]. The simplest alternate model is a uniform pdf. However, a uniform pdf is unrelated to the detailed pdf being estimated. The detailed triphone-dependent pdf can also be combined with less detailed pdfs for the same phoneme [29, 30, 31]; for example models of the phoneme that depend only on left or right context, or a context-independent model. While these models are reasonably related to the detailed model, they are still not as accurate as desired. It is important that the model used to average with must be closely related to the original model, or else the gain in robustness may be offset by having less accurate models.

An alternate method to achieve robustness is to smooth the nonparametric pdfs appropriately. In this section we describe the results of experiments with three different smoothing techniques: *Parzen smoothing, self adaptation cooccurrence smoothing*, and *triphone cooccurrence smoothing*.

In Section 2.1.2 we describe the basic concept shared by all three smoothing techniques, and the detailed algorithms for each of the techniques. Experimental results are given in Section 2.1.3.

2.1.2 PDF Smoothing

The basic tool used by all three smoothing methods is a probabilistic smoothing matrix. This idea was introduced by Sugawara et al. [37] for recognition of isolated digits. Here we apply the method to large vocabulary continuous speech.

For each state of a discrete HMM, we have a discrete probability density function (pdf) defined over a fixed set, N, of spectral templates. For example, in the BYBLOS system we typically use a vector quantization (VQ) codebook of size N =256 [21]. The index of the closest template is referred to below as the VQ index or the spectral bin. We can view the discrete pdf for each state s as a probability row vector

$$\mathbf{p}(s) = [p(k_1\ s),\ p(k_2\ s),\ \cdots,\ p(k_N\ s)], \tag{2.1}$$

where $p(k_i\ s)$ is the probability of spectral template k_i at state s. We can imagine that the probabilities of different spectra are related in that, for each spectrum that has a high probability for a given pdf, there are several other spectra that are also likely to have high probabilities. These might be "nearby" spectra, or they might just be statistically related. We represent this relation by $p(k_j\ k_i)$, the probability that if spectrum k_i occurs, the spectrum k_j will occur also. The set of probabilities $p(k_j\ k_i)$ for all i and j form an $N \times N$ *smoothing matrix*, $\mathbf{T}$, where $\mathbf{T}_{ij} = p(i_j\ k_i)$.

If we multiply the original pdf row vector $\mathbf{p}(s)$ by the smoothing matrix, we get a smoothed pdf row vector.

$$\mathbf{p}_{smooth}(s) = \mathbf{p}_{orig}(s) \times \mathbf{T}. \tag{2.2}$$

In our experiments we use a separate smoothing matrix for each phoneme. This matrix is combined with the phoneme-independent matrix to ensure robustness.

The amount of training available for different models varies considerably, from one or two tokens for the majority of the triphone-dependent models to hundreds of tokens for the more common models. Clearly, we don't want to smooth a model as much if it was estimated from a large number of training tokens. Therefore we *recombine* the smoothed pdf above with the original pdf using a weight $w(s)$ that depends on the number of training tokens of the model. Thus the final pdf used is given by

$$\mathbf{p}_{final}(s) = w(s)\mathbf{p}_{orig}(s) + [1 - w(s)]\mathbf{p}_{smooth}(s). \tag{2.3}$$

The weight w is made proportional to the log of the number of training tokens, N_T:

$$w(s) = \min[0.99, 0.5 \log_{10} N_T(s)]. \tag{2.4}$$

This equation is illustrated in Figure 2.1.

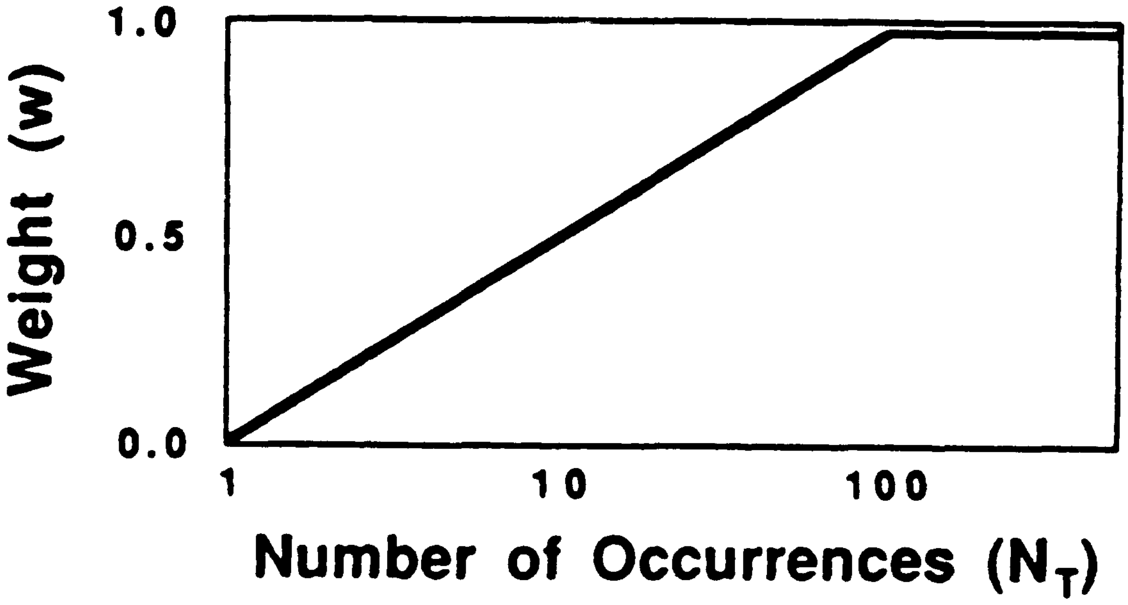


Figure 2.1: Weight w for original model as a function of the number of training tokens, N_T.

We tried three techniques for estimating the smoothing matrix: *Parzen smoothing*, *self adaptation cooccurrence smoothing*, and *triphone cooccurrence smoothing*. Below we describe the three methods.

Method 1: Parzen Smoothing

Parzen estimation assumes that the true probability density varies slowly in space. In other words, points that are close according to some distance metric should also have similar probability densities. We compute a matrix containing the squared Euclidean distance d^2 between each pair of bins. Each distance is then replaced by

$$e^{-(d^2/\sigma^2)^\alpha} \tag{2.5}$$

When α is 1, the function is proportional to a Gaussian. When α is less than 1, it becomes more pointed – for example at $\alpha = 0.5$, it is a Laplacian. When α is greater than 1 it becomes flatter. Next, we normalize each row of the matrix so that it sums to 1.

Method 2: Self-Adaptation Cooccurrence Smoothing

The second method is called *self-adaptation* because it is identical to our method for speaker adaptation [10]. This method is also similar to the "correlation smoothing" method in [37]. Figure 2.2 illustrates the speaker adaptation process. First, we record an additional small set of sentences that have the same text as sentences in the training set. (We used 40 sentences, or about two minutes of speech.)

One of each pair of sentences with the same text is labelled automatically using a decoder with an initial speaker-dependent HMM model, to determine where each phoneme begins and ends. Then, using a standard distance-based DTW algorithm, we align the pairs of sentences. We assume that each aligned pair of frames corresponds to different spectral realizations of the same phonetic event. All the frame vectors are then vector-quantized. For each VQ pair we increment the corresponding two symmetric entries defined by that pair of numbers in the smoothing matrix for the phoneme. Then, each row of the matrix is normalized. The phoneme-dependent matrices are averaged to produce a single phoneme-independent matrix. (Two minutes of speech results in 12,000 pairs of VQ indices.)

Method 3: Triphone Cooccurrence Smoothing

The third method is similar in spirit to the second method, but it tries to overcome two deficiencies. First, it does not require recording any additional sentences. Second, it derives the smoothing matrix from all of the training material rather than just 40 pairs of sentences.

After performing forward-backward training, we have a large number of context-dependent phonetic models. Most of these (about 2,500) are triphone-dependent models. Each model

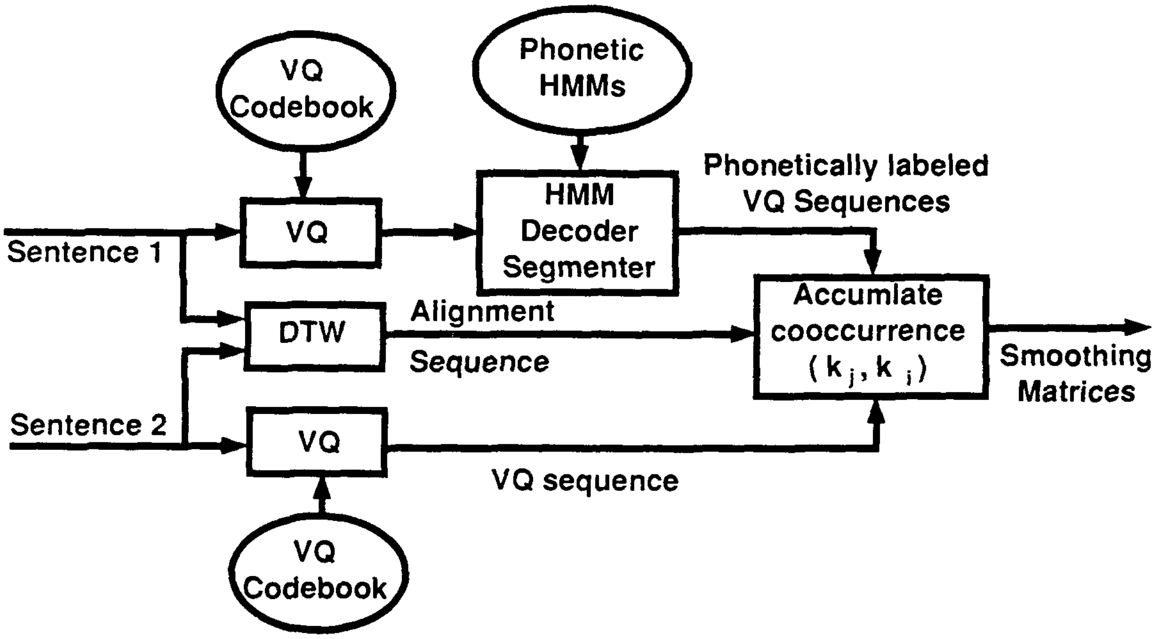
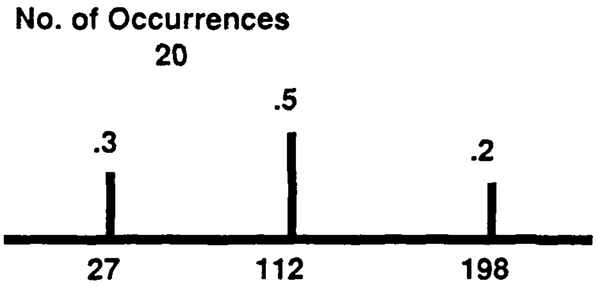


Figure 2.2: Speaker adaptation process. For smoothing, the "target" speaker is just additional speech from the same speaker.



	27	112	198
27	1.80	3.00	1.20
112	3.00	5.00	2.00
198	1.20	2.00	0.80

Figure 2.3: Triphone Cooccurrence Matrix Estimation. pdf shown results in matrix increments shown.

has three different pdfs. Normally we would interpolate these triphone-dependent pdfs with less specific models for recognition. However, before interpolation, these models contain a record of all of the VQ-index spectra that occurred for one part (one state) of a particular triphone. Thus, according to the Markov model, these spectra freely cooccur. For each pdf of each triphone model we count all permutations of two VQ spectra in that pdf, weighted by their probabilities and by the number of training tokens of the model. Figure 2.3 illustrates this process for one pdf of one model.

For example, the pdf shown has VQ indices 27, 112, and 198 with probabilities 0.3, 0.5, and 0.2, respectively. The model occurred 20 times in the training set. Therefore, we add 0.3 * 0.5 * 20 = 3.0 to entries (27,112) and (112,17) in the matrix. As with the second method, we keep a separate matrix for each phoneme and one phoneme-independent matrix. Each row is normalized to create probabilistic matrices. A method similar to this was developed independently by Lee [7]. However, in his method there was only one smoothing matrix, instead of one for each phoneme, and he estimated the matrix from context-independent models instead of triphone-dependent models. We believe that these differences result in too much smoothing.

2.1.3 Experiments

We now describe the experiments in which we compared the performance of the three smoothing methods.

Speech Corpus

The experiments with methods 1 and 3 were performed using the DARPA 1000-Word Resource Management [26] with 600 training sentences (about 30 minutes of speech) for each speaker. Experiments with method 2 were performed using similar material from two speakers recorded at BBN since we needed to repeat sentences; about 350 sentences or 17 minutes of training speech was used. The test material was 25 sentences (about 200 words) for each speaker for all conditions.

Analysis

The speech was lowpassed at 10 kHz and sampled at 20 kHz. 14 Mel-Frequency cepstral coefficients (MFCC) were computed for each 10 ms, using a 20 ms analysis window. At the time these experiments were performed (in early 1988) we only used the steady state cepstral parameters. The effect of the smoothing algorithm on the current system will be discussed at the end of this section.

Training

Five passes of the forward-backward algorithm were used to derive context-dependent models. The smoothing matrices were derived by each method. Then, each context-dependent pdf was smoothed by the appropriate matrix (the matrix for the same phoneme interpolated with the phoneme-independent matrix). For Parzen smoothing there was only one matrix. Then the original and smoothed pdfs were recombined as a given in (2.3). The context-independent phoneme models were never smoothed. Finally, the context-dependent models were combined to construct word models for the entire vocabulary.

Recognition

Recognition experiments were performed with two different language models: the word-pair grammar that is frequently used with this database (perplexity 60) and with no grammar (perplexity 1000). In the original experiments we used the Viterbi algorithm for decoding. The time-synchronous pseudo-Baum-Welch algorithm [30], which gives somewhat better results, was used in the experiments with the current system.

The recognition performance is reported below in terms of total percentage word error rate, defined as

Percent Word Error

Speaker	No Grammar			Word-Pair		
	Interp	Smooth	Smooth + Interp	Interp	Smooth	Smooth + Interp
JWS	25.6	21.8	20.4	4.7	4.7	5.2
PGH	30.0	30.5	29.0	8.5	8.0	5.0
RKM	33.8	35.7	36.2	15.0	16.0	15.5
Average	29.8	29.3	28.5	9.4	9.6	8.5

Table 2.1: Parzen Smoothing. A Gaussian window was used.

$$\%\text{ word error} = 100\ \frac{\text{substitutions} + \text{deletions} + \text{insertions}}{\text{total words}}$$

Results

Table 2.1 contains results for the Parzen smoothing method for three speakers. Separate results are given for interpolated context-dependent models (our standard method), for Parzen smoothing without context-model interpolation, and for both procedures together. As can be seen, the improvements are both small and statistically insignificant.

Table 2.2 shows results for two speakers using method 2: self-adaptation. In this case there is a 24% reduction in error rate for both grammar conditions.

Table 2.3 gives results for 7 speakers for method 3: triphone cooccurrence smoothing. In this case, the reduction in error rate was 10% with no grammar and 30% with a grammar.

Discussion

Of the three methods used, methods 2 and 3, which were based on statistical correlation of VQ spectra worked better than method 1, which was based on a family of spectral distance

Percent Word Error

Speaker	No Grammar			Word-Pair		
	Interp	Smooth	Smooth + Interp	Interp	Smooth	Smooth + Interp
JM	32.2	32.7	25.1	7.6	6.6	5.2
OK	22.2	22.2	16.7	3.4	3.0	3.0
Average	27.2	27.4	20.9	5.5	4.8	4.1

Table 2.2: Self Adaptation Cooccurrence Smoothing. 40 repeated sentences. 2 speakers from BBN.

Percent Word Error

Speaker	No Grammar		Word-Pair	
	Interp	Smooth + Interp	Interp	Smooth + Interp
BEF	35.2	30.5	10.0	6.2
JWS	25.6	21.3	4.7	4.7
PGH	30.0	29.0	8.5	4.5
RKM	33.8	31.5	15.0	11.7
DTD	31.9	27.6	6.2	6.2
DTB	35.0	34.0	9.4	4.9
TAB	22.5	20.3	7.2	4.5
Average	30.6	27.7	8.7	6.1

Table 2.3: Triphone Cooccurrence Smoothing

matrices. This confirms the results in [37]. This does not mean that there is not some distance metric in which a smoothing window would work. Rather the statistical techniques find that metric automatically. The difference between methods 2 and 3 probably comes from the fact that method 3 results in more smoothing than method 2 does. More smoothing is generally more beneficial for recognition with a grammar, and less with no grammar.

2.1.4 Effect of Smoothing on Current BYBLOS System

The original experiments with smoothing were performed two years ago, when the system was different in several ways. In particular, the system now uses three parameter sets of independent codebooks, and models coarticulation between words. Therefore we have measured the impact of our smoothing algorithms on the most recent configurations of the system. The use of between-word phonetic models makes smoothing particularly important because many of the triphones spanning two words occur only once or twice in the training set, and therefore are poorly estimated.

We found that a large number of the triphones that span two words occur only one or two times. These models are not well estimated. However, when we use the triphone cooccurrence smoothing algorithm, the likely bins surrounding the observed bins are "filled in". When we do not use between-word triphones we measure a speaker-dependent word error rate of 3.1% When we use smoothing, this error rate decreases somewhat to 2.7%. If we use between-word triphones without smoothing, the error rate is 2.3% – a 30% reduction from 3.1%. However, when we use smoothing, this error rate drops to 1.6% – or 30% below 2.3%. Thus, the smoothing algorithm is still an essential contributor to the high performance of the system.

As will be seen in later sections, smoothing is also effective for speaker-independent recognition. It is especially effective for cross-speaker recognition.

2.2 Supervised Vector Quantization

In this section we describe several attempts to improve the recognition accuracy with the use of supervised clustering techniques. These techniques modify the distance metric and/or the clustering procedure in a discrete HMM recognition system in an attempt to improve phonetic modeling. We considered three techniques: Linear discriminant analysis, a hierarchical supervised vector quantization technique, and Kohonen's LVQ2 technique. Even though the techniques improved the phonetic recognition capability of the vector quantization, the overall word and sentence recognition accuracy did not improve.

2.2.1 Introduction

Even in a discrete Hidden Markov Model system, there is an underlying distance metric that is used to divide the spectral space into distinct regions. The BYBLOS system currently uses context-dependent phonetic discrete HMMs based on three codebooks. The first codebook contains 14 mel-frequency warped cepstral coefficients (c1-c14) computed every 10 ms directly from the speech power spectrum. The second codebook contains the 14 "differences" of these parameters, derived by computing the slope of a least squares linear fit to a five-frame window centered around each frame [11]. Finally, we use a third codebook that has the amplitude-normalized log rms energy and the "difference" of this energy. We divide these 30 features among three codebooks to avoid the training problem associated with high dimensionality. Each codebook is designed using a nonuniform binary clustering algorithm, followed by several iterations of the k-means algorithm [21]. The k-means clustering algorithm uses Euclidean distance.

As a possible method for improving recognition accuracy, we investigated the use of linear discriminant analysis [6, 8]. We also considered several methods of nonlinearly warping the spectral space as part of the vector quantization process. We call these methods "supervised clustering" techniques. To use these techniques, we need to define the classes that we want to discriminate. We chose the (50 or so) basic phonemes as that set, under the assumption that these represent most of the distinctions that must be made in large vocabulary speech recognition. To obtain phoneme labels for the training data we first estimate speech models using the standard techniques in the BYBLOS system and then segment automatically all of the training data into phonemes using the decoder (recognizer), constrained to find the correct answer. The recognized segment boundaries are then used to assign a phoneme label to each frame. Each of the techniques described below then attempts to define a distance metric or vector quantizer that can recognize the phoneme label of a single frame of speech.

2.2.2 Linear Discriminant Analysis

Brown [6] has proposed using several successive frames jointly in order to model the joint density of the observed speech more accurately. He then uses linear discriminant analysis (LDA) to reduce the number of dimensions. We attempted to use LDA on our 30 mixed features to find a set of features that would, in fact, be more independent. In addition, we hoped that we would automatically find a more beneficial weighting of the different features than simple Euclidean distance.

We compute the within (phoneme) class and between class means and covariances of all the frames of training data. We use the generalized eigenvector solution to find the best set of linear discriminant features. Then, we simply cluster and quantize the 30 new features as usual. Alternatively, we can divide the new features up into a small number of codebooks in order to reduce the quantization error. We use these new (quantized) features in place of the original features for discrete HMM continuous speech recognition.

2.2.3 Supervised Vector Quantization

In addition to simple linear discriminants, we consider more complex warpings of the feature space. We call the general approach *supervised clustering* or *supervised VQ*. Instead of finding a codebook that minimizes mean square error, without regard to phonetic similarity, we use the training data to generate a codebook that tends to preserve differences that are phonetically important, and disregard feature differences (even if they are large) that are not phonetically important. In effect, we attempt to maximize the mutual information between the VQ clusters and phonetic labels. We describe two techniques below that seem suitable for accomplishing this goal.

Binary Division of Space

The first algorithm is most closely related to the nonuniform binary clustering algorithm that we use to derive an initial estimate for k-means clustering [21]. All the labeled frames are initially placed in one cluster. Then, we iteratively divide the clusters until we have the desired number. One of the many clustering algorithms we tried is given below.

First we measure the entropy reduction that would result from dividing a single cluster into two:

1. Estimate a single diagonal-covariance Gaussian for the frames with each phoneme label in the cluster.
2. Identify the two most "prominent" phonemes within the cluster.
3. Divide all the frames in the cluster into two new clusters using these two Gaussian distributions.

4. Compute the difference between the entropy of the phoneme labels in the original cluster and the average entropy of the two new clusters, weighted by the number of samples in each subcluster.

The outer loop repeatedly divides the cluster that will result in the largest enropy reduction.

1. Divide the cluster that would result in the largest entropy reduction.
2. Create two new clusters and measure the potential entropy reduction for dividing each of the two resulting clusters as described above.
3. If we have fewer than 256 clusters, go to (1).

The one-step lookahead avoids dividing a large cluster when no reduction in entropy would result. The resulting codebook is then used to quantize all of the training and test data. While this algorithm increased the mutual information between the codebook and the phonetic labels, there was no gain in the overall recognition accuracy.

LVQ2: Kohonen's Learning Vector Quantizer

The LVQ2 algorithm [17] was used very effectively in a phoneme recognition system. [22]. The algorithm amounts to a discriminative training of the codebook's means to maximize recognition of the frame labels.

As before, we start with the set of phonetically labeled frames. For LVQ2, we use a "sliding window" of some fixed size centered around each frame to create large feature vectors (7-frame windows were used in [22]). Then we use the binary and k-means algorithm to divide the feature vectors from each phoneme into several clusters. We make the number of clusters for each phoneme proportional to the square root of the number of frames of that phoneme, such that the total number of clusters is 256. Each cluster has the name of the phoneme data in it. Then, we use LVQ2 to shift the cluster means to optimize frame recognition. We review the algorithm below briefly. For each feature vector:

1. Find the nearest cluster and the next nearest cluster from a different phoneme.
2. If the nearest cluster is from the wrong phoneme and the second nearest is of the correct phoneme, shift the mean of the correct cluster toward the feature vector in question and shift the wrong cluster's mean away, according to:

$$\mathbf{m}_i(t + 1) = \mathbf{m}_i(t) - \alpha(t)\,(x(t) - \mathbf{m}_i(t))$$

$$\mathbf{m}_j(t + 1) = \mathbf{m}_j(t) + \alpha(t)\,(\mathbf{x}(t) - \mathbf{m}_j(t))$$

where $\mathbf{x}$ is a training vector belonging to class j,
$\mathbf{m}_i$ is the reference vector for the incorrect category,
$\mathbf{m}_j$ is the reference vector for the correct category, and
$\alpha(t)$ is a monotonically decreasing function of time.

The above algorithm is iterated until convergence (which requires some care). As suggested in the reference, we used several adjacent speech frames together as a longer feature vector. We also performed experiments in which we used the LVQ2 algorithm separately on several frames of steady state cepstra and several frames of difference cepstra. The resulting codebooks are used in the normal way.

2.2.4 Experiments And Results

We performed speaker-dependent recognition experiments on a six-speaker subset of the DARPA Resource Management speech corpus, using the May 1988 test set. Continuous speech recognition experiments were done using the word-pair grammar and also with no grammar.

To understand the behavior of the supervised clustering algorithms, we measure the correspondence between the resulting codebooks and the phonetic labels. First, we determine the most frequently occurring phoneme label within each cluster. We then quantize new frames and use the VQ indices to "recognize" the phoneme labels of the independent data. Table 2.4 shows the phoneme-frame recognition accuracy for training and test data, and for steady state and difference cepstra. The results show that codebooks made by the binary split algorithm are slightly better than the unsupervised k-means algorithm at recognizing a phoneme label from a single frame of steady-state cepstra. We performed similar frame recognition experiments using LVQ2 with sliding windows of length 1, 3, 5, and 7 frames. Table 2.4 shows that LVQ2 is better than both k-means and binary division at predicting frame phoneme labels, even when LVQ2 does not look at a frame's neighbors. As we increase the window size, the correspondence between the VQ clusters and phonemes improves significantly.

The results of continuous speech recognition experiments are given in Table 2.5. The control experiment for these results which used a somewhat older version of the BYBLOS system is labeled k-means. The HMM recognition system has a number of system parameters. Wherever possible, these parameters are left unchanged between the k-means control and the other tests.

Metric /		Cepstra		Diff Cepstra	
Algorithm		train	test	train	test
K-means	1 frm	43%	41 %	23 %	21 %
Binary	1 frm	45	42	25	22
LVQ2	1 frm	49	44	29	25
LVQ2	3 frm	57	51	39	33
LVQ2	5 frm	62	55	46	39
LVQ2	7 frm	65	57	50	43

Table 2.4: Frame Phoneme Recognition Rate

The last condition in Table 2.5 (labeled "Recent BBN") corresponds to the most recently reported performance of the BBN BYBLOS system for this subset of the May 1988 data [1]. The system that produced this performance is similar to the control system, except that in addition to modeling coarticulation within words, we used cross-word context-dependent phonetic models (triphones) to model coarticulation between words. This experimental result is included purely for reference.

We discuss four LDA experiments with variations in the number of codebooks and assignment of linear discriminants to codebooks. In all four tests we concatenated the 14 cepstral coefficients, the 14 "difference" coefficients, and the two normalized energy coefficients, and used LDA to extract a new set of 30 discriminant features. In the first test (30f) we then clustered all 30 discriminant features into one codebook which was used in HMM recognition. In the second test (15f,15f), we split the 30 discriminant features into two 15-parameter codebooks. In the third test (15f) we used only the first 15 discriminant features in a single codebook. Finally, in the fourth test (km,15f), we used the standard three codebooks together with a fourth codebook containing the first 15 discriminant features. As can be seen in Table 2.5 most results using LDA did not improve over the baseline 3-codebook condition.

We performed three recognition tests of supervised clustering using the binary division algorithm. In the first test (30f), we concatenated all 30 parameters into a single feature vector and created one codebook using binary division. In the second test (c,d,e), we used binary division to cluster separately the cepstrum, difference cepstrum, and energy coefficients. This three-codebook experiment, then, is a direct comparison between binary division and unsupervised K-means. As Table 2.5 shows, neither the one-codebook nor the three-codebook binary division experiments resulted in improved recognition over the baseline. As a third test (km,30f), we add the 30-feature binary-division codebook from the first test to the three unsupervised codebooks of the control. This four codebook experiment

System	Grammar		
	Codebk(s)	word-pair	none
K-means (km)	c,d,e	3.6 %	18.8 %
Lin Discrim	30f	5.1	20.3
Lin Discrim	15f,15f	4.7	20.9
Lin Discrim	15f	6.2	24.9
Lin Discrim	km,15f	3.8	16.1
Binary Div	30f	4.8	20.6
Binary Div	c,d,e	3.9	17.2
Binary Div	km,30f	3.3	17.1
LVQ2 (1 frm)	c,d,e	4.0	18.5
LVQ2 (3 frm)	c,d,e	4.2	17.7
LVQ2 (5 frm)	c,d,e	3.3	18.1
LVQ2 (7 frm)	c,d,e	3.9	18.6
LVQ2 (3/5 frm)	c,d,e	3.2	18.1
Recent BBN	c,d,e	2.1	13 7

c = cep, d = dif, e = energy, <#>f = <#> features

Table 2.5: Word Recognition Error: Multiple codebooks

results in a small (10%) reduction in error rate relative to the three codebooks by themselves.

Next, we show several three-codebook recognition experiments using the LVQ2 algorithm with windows of size 1, 3, 5, and 7 frames on the cepstral and difference coefficients. For simplicity, the energy parameters were clustered using the baseline K-means algorithm. In the last experiment, the cepstra codebook uses a 3 frame window and the difference-cepstra codebook uses a 5 frame window. While there are small random variations in the results, there are no significant improvements in overall recognition accuracy.

We were surprised that the LVQ2 algorithm improved the frame recognition accuracy so much without improving the overall speech recognition accuracy. Therefore, we performed an additional set of experiments using only one codebook with steady-state cepstra. These results are summarized in Table 2.6 . We see, again, that the improvement in cluster/phoneme correspondence from increasing LVQ2's window size does not necessarily translate into better system word recognition.

System (cepstra–only)		Grammar		frame
		word-pair	none	recog
K-means	(1 frm)	8.4 %	30.0 %	41 %
LVQ2	3 frm	7.5	30.8	51
LVQ2	5 frm	7.5	28.9	55
LVQ2	7 frm	8.2	29.8	57

Table 2.6: Word Recognition Error vs Frame Recognition Accuracy for one codebook

2.2.5 Discussion of Results

The results generally show that, even when the supervised clustering is successful at improving the correspondence between the VQ codebook regions and the phonetic labels, the overall speech recognition accuracy does not improve. We can draw two possible conclusions from these results relative to previous successes with these techniques. First, while it might be possible to find a small number of discriminant directions that are important for a small vocabulary task – especially one with minimal pair differences – it may not be as easy in a large vocabulary task, where the important distinctions are many and also very varied. That is, any choice of discriminants that is better for some distinctions may be worse for others. Second, it is not clear that optimizing phonetic distinctions on single frames will help a recognition system whose goal is to recognize words using triphone models.

2.3 MMI Estimation of Codebook Weights

In the BYBLOS system we currently use three independent parameter sets to represent speech. We multiply the discrete probability of the three sets, assuming independence. However, there is no reason to believe that these three sets are equally useful. We have found empirically that we can improve recognition accuracy by giving more weight to some codebooks. This is accomplished by having the probability for each set exponentiated by a corresponding weight.

We would like to be able to estimate these weights automatically in order that they could vary across different speakers. However, these exponential weights can not be estimated using maximum likelihood since the likelihood would be maximized when all the exponents are set to 0. If the exponents were constrained to sum to 1, maximum likelihood would set one of them to 1 and the rest to 0. Therefore, we investigated the use of Maximum Mutual

Information techniques to estimate the exponential codebook weights.

2.3.1 MMI Estimation for Continuous Speech

In maximum likelihood estimation, we want to find a parameter vector $\hat{\theta}$ so that the quantity

$$Pr_{\hat{\theta}}(A, W) = \max_{\theta} Pr_{\theta}(A, W)$$

is maximized, where $Pr_{\theta}(A, W)$ is the joint probability of uttering text W and producing acoustic evidence A, using a model parameterized by θ. Furthermore, we can factor $Pr_{\theta}(A, W)$ as

$$Pr_{\theta}(A, W) = Pr_{\theta}(A \mid W)\, Pr_{\theta}(W)$$

where the *acoustic model* $Pr_{\theta}(A \mid W)$ and the *language model* $Pr_{\theta}(W)$ can be estimated separately.

Maximum Mutual Information Estimation: It has been shown that ML will give the optimal estimator under a certain set of conditions [23]. However, these conditions are rarely if ever satisfied in speech recognition. One alternative, as proposed in [6], is to use maximum mutual information parameter estimation. Instead of choosing parameter vectors θ to maximize the joint likelihood $Pr_{\theta}(A, W)$, the objective function in MMI is the *mutual information* between the events A and W:

$$\begin{aligned} \mathrm{I}_{\theta}(W; A) &= \log \frac{Pr_{\theta}(A, W)}{Pr_{\theta}(A)\, Pr_{\theta}(W)} \\ &= \log Pr_{\theta}(A \mid W) - \log Pr_{\theta}(A) \qquad (2.6) \end{aligned}$$

In MMI, the parameter vector θ is chosen to maximize $\mathrm{I}_{\theta}(W; A)$. $Pr_{\theta}(A)$ in (2.6) can be further expanded as

$$Pr_{\theta}(A) = \sum_{\hat{W}} Pr_{\theta}(A \mid \hat{W}) Pr_{\theta}(\hat{W}). \qquad (2.7)$$

As can be seen, the term $\log Pr_{\theta}(A)$ in the criterion implicitly takes all possible word sequences $\hat{W}$ into account.

Assuming that the language model $Pr_{\theta}(W) = Pr(W)$ is given, the goal in MMI is to choose acoustic parameters θ to maximize information. Taking the derivative of $\mathrm{I}_{\theta}(W; A)$ with respect to parameter θ_i

$$\frac{\delta \mathrm{I}_\theta(W; A)}{\delta\theta_i} = \frac{\frac{\delta Pr_\theta(A|W)}{\delta\theta_i}}{Pr_\theta(A \mid W)} - \frac{\sum_{\hat{W}} \frac{\delta Pr_\theta(A|\hat{W})}{\delta\theta_i} Pr(\hat{W})}{Pr_\theta(A)} \tag{2.8}$$

The first term in (2.8) corresponds to the derivative of the maximum likelihood objective function. The second is an additional term introduced by maximum mutual information, which subtracts a component in the direction of $\frac{\delta Pr_\theta(A|\hat{W})}{\delta\theta_i}$ for each possible word sequence $\hat{W}$ in the language, including W. In ML, the objective is to increase $Pr_\theta(A \mid W)$ for the correct word sequence W. In MMI, the objective is to increase $Pr_\theta(A \mid W)$, but also to decrease $Pr_\theta(A \mid \hat{W})$ for every *incorrect* word sequence $\hat{W}$. This is the major difference between ML and MMI.

Applying (2.8) to an HMM, for the case of θ being the parameters of an output distribution,

$$\frac{\delta Pr_\theta(A \mid \hat{W})}{\delta\theta_j} = \sum_{t=1}^{T} \sum_i \alpha_i(t-1) a_{i_j} \left(\frac{\delta b_j(Y(t))}{\delta\theta_j}\right) \beta_j(t) \tag{2.9}$$

and

$$\frac{\frac{\delta Pr_\theta(A|\hat{W})}{\delta\theta_j}}{Pr_\theta(A \mid \hat{W})} = \frac{\sum_{t=1}^{T} \sum_i \alpha_i(t-1) a_{ij} \left(\frac{\delta b_j(Y(t))}{\delta\theta_j}\right) \beta_j(t)}{\sum_i \alpha_i(T)} \tag{2.10}$$

$$= \sum_{t=1}^{T} \gamma_j(t) \left(\frac{\frac{\delta b_j(Y(t))}{\delta\theta_j}}{b_j(Y(t))}\right) \tag{2.11}$$

where $\alpha_i(t-1)$ and $\beta_j(t)$ are the forward and backward probabilities, respectively, in the Baum-Welch computation, $\gamma_j(t)$ is the conditional probability of state j at time t used in reestimation, and a_{ij} and $b_j(Y(t))$ are the transition and output probabilities, respectively, where the output distributions are associated with the state (for the sake of simplicity, a^nd without loss of generality, the tying of distributions is not treated here, so each state is assigned a unique distribution).

2.3.2 MMI Estimation of Exponential Coefficients

In the BYBLOS system, the input features in general are represented as multiple streams of discrete labels produced by multiple vector quantization (VQ) codebooks (corresponding to decomposition of input features into multiple feature sets), so that for each 10 millisecond frame interval, N output symbols are observed [13]. We make the assumption that the labels at a frame are independent, so that the joint label probability density function (pdf) at the state can be written as a product of the individual densities. We can assign exponential weighting coefficients to the individual pdfs to reflect the relative importance of the various feature sets for recognition. In addition, we use a robust context-dependent phonetic modeling technique to model the output pdf at a state in a word [30], so that the pdf for a state is computed as an interpolated pdf of M context pdfs (we typically use context-independent, left, right, and triphone contexts). We express the conditional pdf at state j as

$$\begin{aligned} b_j(Y(t)) &= \sum_{m=1}^{M} \lambda_m b_{jm}(Y(t)) \\ &= \sum_{m=1}^{M} \lambda_m \prod_{n=1}^{N} b_{jmn}^{\zeta_{mn}}(y_n(t)) \end{aligned} \qquad (2.12)$$

where $b_{jm}(Y(t))$ is the conditional probability at state j of observing the vector of N labels $Y(t)$ at time t for context model m, λ_m the prior probability (context weight) for context pdf m, $b_{jmn}(y_n(t))$ the conditional probability of observing the nth output label $y_n(t)$ at time t for context pdf m, and ζ_{mn} the exponential weighting coefficient assigned to distribution b_{jmn}.

Previously, the assignment of the coefficients ζ_{mn} was done by trial and error: their values are chosen empirically to give good recognition performance, and these values are fixed across all test speakers. We would like to be able to estimate the ζ_{mn}'s automatically from data in order for them to be speaker-specific, and also let them vary as a function of the context model m, to improve recognition accuracy. Since ML methods would invariably choose only one codebook (one with highest average probability on data) and set the rest to zero, the estimation of codebook weights can only be done with MMI-like training paradigms.

With this goal in mind, we can apply (2.11) and using the particular form of the distribution give by Equation (2.12), with θ being ζ_{mn} for a particular codebook n and a particular context model m,

$$\frac{\frac{\delta b_j(Y(t))}{\delta \zeta_{mn}}}{b_j(Y(t))} = \frac{\lambda_m \log(b_{jmn}) \prod_{l=1}^{N} b_{jml}^{\zeta_{ml}}(y_l(t))}{b_j(Y(t))}. \qquad (2.13)$$

Taking(2.11), and summing over states j, we have

$$\frac{\frac{\delta Pr_\theta(A|\hat{W})}{\delta \zeta_{mn}}}{Pr_\theta(A \mid \hat{W})} = \sum_j \sum_{t=1}^{T} \gamma_j(t) \frac{\lambda_m \log(b_{jmn}) \prod_{l=1}^{N} b_{jml}^{\zeta_{ml}}(y_l(t))}{b_j(Y(t))}. \qquad (2.14)$$

In the case where we allow ζ to vary only with n, we simply take the sum over m of the numerator, and

$$\frac{\frac{\delta Pr_\theta(A|\hat{W})}{\delta\zeta_n}}{Pr_\theta(A \mid \hat{W})} = \sum_j \sum_{t=1}^{T} \gamma_j(t) \frac{\sum_{m=1}^{M} \lambda_m \log(b_{jmn}) \prod_{l=1}^{N} b_{jml}^{\zeta_l}(y_l(t))}{b_j(Y(t))}. \tag{2.15}$$

2.3.3 Implementation Issues

Finding the Imposter Sentences: In computing the MMI objective function and its associated maximization, one needs to compute the probabilities $Pr(A \mid \hat{W})$ for all $\hat{W}$. Of course this is practically infeasible for any reasonable size vocabulary. In practice, one only needs to compute $Pr(A \mid \hat{W})$ for a (relatively small) subset of $\hat{W}$, those which are confusable with the correct word sequence W. Bahl et al. [2] derived the imposter $\hat{W}$'s in a 2000-word isolated word task by performing an acoustic fast match for each word in the utterance, and the $\hat{W}$'s were taken to be those words that were not pruned by the fast matcher.

For a continuous speech recognition task, we compute the $\hat{W}$'s using the N-best algorithm, which computes the N best scoring complete sentence hypotheses for each utterance [35]. This algorithm is computationally efficient, and its output can be used directly in MMI training. For the results reported in this paper, we limit the number of imposter hypotheses per utterance to 10, so that the N-best alternatives can be computed relatively quickly. In fact, since we are using a perplexity 60 finite-state grammar, it was almost always the case that the top 10 alternatives account for most of the likelihood $Pr_\theta(A)$.

Maximizing the Objective: The hill-climbing technique that we use to maximize mutual information $I_\theta(A; W)$ is a gradient descent where we first compute the gradient (from the $\gamma_i(t)$'s) of the objective function, and perform a line search along the gradient to optimize the objective function, and then repeat. This is an iterative procedure to which the following heuristics are incorporated to minimize computation:

1. To minimize the computation during line search, which in general requires computing the objective function (the α's) several times, we use a hybrid method where the line search is terminated immediately upon finding a better solution than the current one. This ensures that the gradient descent always gets to a better point in the solution space, and yet can be more ambitious in taking a larger step for faster convergence (in general, the straightforward descent algorithm is guaranteed to increase the objective function if a small enough step is taken in the direction of the gradient, but it may require a large number of iterations to get to the maxima).

2. In addition, to get a reasonable initial step size for the line search and make this descent algorithm better behaved for this problem, at each iteration we limit the initial step size Δ_i to a percentage ω of its value:

$$\Delta_i \leq \frac{\omega \zeta}{\left| \frac{\delta \mathbf{l}_\theta(A:W)}{\delta \zeta} \right|}$$

3. Finally, since we are only interested in finding a reasonably good solution, and not necessarily the globally best solution, we limit the number of iterations of gradient descent. In practice, we found that since the solution space is highly constrained (with only a few degrees of freedom), a near-optimal solution is almost always reached within five iterations.

2.3.4 Experimental Results

In this section, we present recognition results using MMI for estimating the exponential codebook coefficients, on the standard DARPA 1000-word Resource Management speaker-dependent speech corpus [26], using the Word Pair Grammar (perplexity = 60). Input speech was sampled at 20 kHz, and 14 Mel-Frequency cepstral coefficients (MFCC), their derivatives (dMFCC), plus power (R0) and its derivative (dR0) were computed for each 10 ms, using a 20 ms analysis window. Three separate 8-bit codebooks were created for each of the three sets of features using K-means vector quantization.

In these experiments, context-dependent acoustic models were trained using 600 training sentences (about 30 minutes) for each speaker. An additional 100 sentences were used to estimate the codebook coefficients: first, the N-best alternatives are computed; these alternatives are then used in estimating the weighting coefficients for the different codebooks in the HMM using gradient descent (see Equations 2.8 and 2.11). In all of the results, five iterations of gradient descent were used. An independent set of 25 sentences per speaker were used for testing.

Table 2.7 shows the recognition results [1] averaged across 6 speakers of applying MMI estimation to exponential codebook coefficients. In Experiment C_3, 3 coefficients were estimated, and in C_{12}, 12 coefficients (context-specific). As can be seen, estimation of codebook weights C_3 achieved a modest reduction of 11% in word error rate (3.2% vs 3.6%) over the baseline. However, allowing the coefficients to be context specific (C_{12}) did not improve performance (3.4% word error).

[1] The baseline results shown here do not reflect the most up-to-date system. This experiment was used strictly and only for comparison of techniques.

Method	Word Error	Sent Error
Baseline	3.6	19.3
C_3	3.2	20.0
C_{12}	3.4	19.3

Table ?.7: Recognition results using Word Pair Grammar (perplexity = 60).

	Codebook Coefficients			Mutual Info Gain
Speaker	MFCC	dMFCC	R0 + dR0	(bits)
bef	.39	.42	.04	+0.22
cmr	.43	.60	.11	+1.76
das	.18	.53	.37	+0.15
dtb	.36	.60	.11	+0.39
dtd	.40	.63	.11	+0.37
ers	.41	.54	.11	+0.86

Table 2.8: Statistics of MMI training of codebook weights (5 iterations of gradient descent). Initial weights = (.50, .67, .33)

Table 2.8 shows the resulting codebook coefficients and mutual information gain as a function of the speaker, after MMI training with 5 iterations of gradient descent. As can be seen, the coefficients have changed signficantly from the initial values of .5, .67 and .33, and the amount of change varies depending on the speaker. Also the mutual information improved consistently across all the speakers.

For experiment C_{12}, larger gains in mutual information were observed as a result of training. Again, the coefficients had changed noticeably from their initial values. Also, the derived coefficients were different for different contexts, indicating the utility of context-specific coefficients, although the recognition results didn't seem to reflect that.

Although the results demonstrated so far are only moderately positive, it may in fac· be the case that perhaps it is to be expected. The initial coefficients used in the estimation had already been tuned previously to optimize recognition performance, so that MMI estimation only serves to fine tune them further, for each speaker. The discriminative effect of a few exponential parameters is quite limited. However, further work is needed to fully explore this powerful technique of parameter estimation for speech recognition.

2.3.5 Conclusions

In summary, we presented a useful application of MMI estimation to a particular type of HMM parameters, namely, exponential codebook parameters, in continuous speech recognition. The MMI computation was made feasible by making use of the N-best decoding algorithm for computing the alternate sentence hypotheses used in maximizing the objective function. Although the results were only moderately positive, it does demonstrate the utility of this technique. Further work is needed not only for the particular problem reported, but for estimating other parameters of interest in speech recognition.

2.4 Ear-Model Signal Processing

We implemented the "ear model" signal processing algorithm described by Jordan Cohen of IBM. This signal processing method computes a filter bank model from the power spectrum and normalizes the filter levels dynamically. It was reported to reduce the sensitivity of the IBM system to differences in channel which result from using a desk-mounted microphone. When we used this signal representation as an alternative to our standard cepstral analysis we found no improvement in the recognition accuracy. Perhaps this is because the signal we are using is quite clean and has little variability.

2.5 Phonetic HMM Topology

All of the experiments at all of the DARPA sites using HMM techniques have used the simple 3-state phonetic model that we introduced several years ago. To create more detailed models, we experimented with larger, more detailed phoneme topologies. For example, 5 or 13 states for each phoneme. We also experimented with different HMM phoneme topologies for different phonemes. For example, phonemes that were likely to be long had longer topologies, while phonemes that were often short had shorter topologies. Surprisingly, the initial experiments indicated that using the basic 3-state model for all phonemes yields the best performance.

2.6 Modeling Coarticulation Between Words

Most of the recognition errors in the system involve short words. These words are largely affected by coarticulation from neighboring words. Therefore, we have extended the basic context-dependent modeling to model the dependency of the acoustics of the first and last phonemes in a word on the identity of the adjacent phonemes in the neighboring words. We implemented a simple way of modeling this dependence without changing most of our software. Basically, given a phonetic vocabulary and a grammar, we redefine both so that we end up with a grammar of phoneme models that embody both the desired word sequence constraints and also include the coarticulation effects across word boundaries. The result was that the error rate was decreased by 30% under most conditions. Interestingly, since there are now a large number of models that are estimated from only one or two training samples, it became more important that the models be smoothed using our cooccurrence smoothing algorithm. When the smoothing algorithm was not used, the gains from using between-word coarticulation was cut in half.

2.7 Deleted Estimation of Context Weights

The BYBLOS system interpolates all the different probability densities of the context-dependent phonemes to obtain a robust estimate of the densities. Currently we use heuristic weights that are a function of:

- type of context (phone, left, right, triphone)
- number of occurrences in training (5 ranges)
- state in phone model (left, middle, right)

The values of these weights were set based on reasonable intuitions about the importance of phonetic contexts and amount of training on different parts of a phoneme. We ran a few tuning experiments (on an earlier database) to determine rough scaling factors on the initial weights. Therefore, it is likely that we would see no further improvement by estimating the weights automatically with deleted estimation. However, we might expect that if we estimated the weights automatically, we could use different weights for each speaker. We wanted to avoid any approximations if possible, due to assumptions about the alignments remaining fixed, and so we chose to iteratively estimate the weights and then reestimate the probability densities.

We were worried about the effectiveness of the jackknifing procedure that is normally used, since the weights for combining models are estimated for the case where only half of the data was used to estimate the models. Therefore, we developed a method for holding out only one utterance at a time, that was still very efficient.

Each normal pass of forward-backward is followed by a second pass that estimates the weights. At the end of the forward-backward pass, we retain the "counts". In the second pass we remove the "counts" from one sentence at a time and then estimate context weights using that deleted sentence. The procedure follows:

1. Run usual forward-backward iteration on all sentences.
2. For each sentence:
   (a) Run forward-backward on this sentence using "old" model to determine its contribution to the new model.
   (b) Subtract the contribution of this sentence from those models relevant to this sentence.
   (c) Run forward-backward to compute weight counts from this sentence using the model with the contribution for this sentence removed.
3. Reestimate the context weights from the weight counts.
4. iterate

This algorithm requires only two times the computation of the normal forward-backward algorithm, and should result in a more accurate estimate of the weights than the usual procedure. Unfortunately, when we ran our initial experiments, we found no improvement, despite the fact that the likelihood of the training data had increased somewhat. It is possible that the initial heuristic weights are close enough, or that the "reasonable" continuity constraints existing in the initial weights were lost when each weight was estimated independently.

Chapter 3

Speaker-Independent Modeling

Our work in speaker-independent recognition falls into two categories. First, we performed experiments using the DARPA Resource Management 109-speaker corpus; using various techniques we have reduced the error rate for this scenario to 3.9%, which is the lowest error rate reported to date for this corpus. The second category involves a new paradigm for speaker independent training. Rather than requiring a small amount of speech from a large number of speakers, we use a larger amount of speech from a relatively small number of speakers, which is a much more practical scenario. In addition, we found that a simplification to the training paradigm also improved the results.

3.1 Improvements to Speaker-Independent Recognition

We had previously experimented with the second derivatives of the spectral parameters as additional features for speaker-dependent recognition, with no significant improvement. We also had experimented with tied-mixture densities and a greater number of smaller codebooks. Most recently, we reported improved speaker-independent recognition simply by training the male and female speakers separately. We combined all of these system features and ran a series of experiments on speaker-independent recognition.

The basic 3-codebook result for the 109-speaker training set using the Word-Pair Grammar was 6.5% word error. When we used separate male and female models, the error rate went down to 5.5%. When we added codebooks with the second derivative of the cepstrum and power, the error rate went down to 4.9%. And finally, when we used the tied-mixture representation of the data to estimate the models, the error rate decreased to 3.9%. This error rate is the lowest of any reported thus far using this training set. (For comparison, the result reported for the CMU Sphinx system at the June 1990 meeting was 4.6%.)

3.2 New Paradigm for Speaker-Independent Training

It is a widely held belief that speech used for training SI models must be collected from many speakers. It is also commonly accepted that collecting only a small sample of speech from each training speaker is a reasonable compromise to make in the effort to collect as many speakers as possible. While this compromise may be reasonable for SI recognition, several efforts to use such a corpus as a basis for speaker adaptation have failed to make significant improvements.

Recently, we have discovered that adequate SI performance can be achieved with far less speaker coverage than conventionally thought necessary, but with much better sampling of each training speaker's speech. Specifically, we show that it is possible to achieve near state-of-the-art SI performance on a 1000-word continuous speech recognition task using only 12 training speakers. Furthermore, we will show that it is possible and advantageous to create the SI model from a set of independently trained speaker-dependent (SD) models, without retraining on the entire pooled dataset at one time. Most importantly, we show in Chapter 4 that such a SI corpus is an effective basis for speaker adaptation.

SI Training with Few Speakers

It would be far preferable if we could train a SI system using large amounts of speech from a few speakers. First, in many cases it is inconvenient to arrange for, set up, and collect a few sentences from a large number of speakers. It is clearly much easier and faster to collect a large amount of speech from a few speakers. Second, in contrast to the usual scenario, there is a very large incentive for those speakers who are the training speakers. In our recent experiments, we have seen that the recognition accuracy for the training speakers is almost the same as speaker-dependent performance, i.e., two to three times lower error rate than SI performance.

Below we describe a series of experiments in which we used the speech from the 12 speakers in the SD portion of the DARPA corpus. The training for each speaker consisted of 600 training utterances. Seven of the speakers are male and five are female. The experiments explored many ways of combining the speech from different speakers and considered the most effective method for using our robust smoothing techniques.

Independent Smoothing and Training

We did have some indication that pooling the data of even a few speakers could make large improvements from an experiment conducted at IBM and described in [28]. However, 12 speakers could hardly be expected to contain an example of all speaker types in the general population (including both genders), so we could anticipate the need for some kind of smoothing before we began. Our usual technique for smoothing across the bins of the discrete densities, triphone cooccurrence smoothing [34], has proven to be an effective method for dealing with the widely varying amounts of training data for the detailed context models in the system. When used in a SD training scenario, it has allowed us to observe a performance gain for explicitly modeling several thousand triphones which were observed only once or twice in the training.

However, the cooccurrence smoothing is not appropriate for models derived from the pooled data of many speakers. Spectra from different speakers will cooccur much more randomly than spectra from a single speaker. This will yield poorer estimates of the smoothing matrices. As such, triphone cooccurrence smoothing is a *speaker-specific* modeling technique. If the data is pooled prior to training, we cannot effectively apply our best smoothing to the model.

This realization has led us to examine the practice of pooling the data in the first place. A straightforward alternative to pooling the data is to keep the speakers separated until the speaker-specific operations of training and smoothing have been completed and then combine the multiple SD models. To allow the model combination to be done by averaging the model statistics, we constructed a SI codebook which was used in common for all speakers.

Results of SI Experiments

Results for several SI experiment are shown in Table 3.1. All results are from first runs of the designated Feb. '89 SI test set on the given system configuration. This test set consists of 10 speakers (4 females) with 30 utterances each. All runs used the standard word-pair grammar of perplexity 60. System parameters were fixed before running any of the conditions in this experiment. The limited development testing which we did perform was done only on the June '88 SD/SI test set using only the 109 speaker SI model.

For each condition we show the number of training speakers, and the manner in which the models were trained and smoothed. The training was done either on pooled data (*joint* training) or on individual speakers' data (*indep* training). The smoothing was either not done, or was applied to either the jointly or independently trained model. For each condition, the word error rate (which includes insertion errors) and sentence error rate are given.

#Spkrs	Training	Smoothing	Word Err	Sent Err
109	joint	none	7.1	36
109	joint	joint	6.5	34
12	joint	none	9.0	42
12	joint	joint	8.5	41
12	joint	indep	7.8	37
12	indep	indep	7.5	37

Table 3.1: Comparison of SI training scenarios on the Feb. '89 test set with word-pair grammar.

The 109 speaker conditions were run to calibrate the BYBLOS system with published results for the same test set. We observe a small improvement, from 7.1% to 6.5% word error, for using smoothing on the jointly trained model. The 6.5% error rate is comparable to the best performance on record (6.1%) for this test set which was achieved by Lee as noted in [20]. Furthermore, the sentence error rates are identical. Lee's system used a corrective training and reinforcement procedure to increase the discrimination ability of the model for confusable words. No corrective training was used for the BYBLOS results given in Table 3.1.

The system configuration for the 109 speaker condition was identical to that which we use for SD recognition except for one difference. One new system parameter was added to decrease the factors used for combining the context-dependent models into interpolated triphones [30] by a factor of eight to account for the larger corpus.

Next we repeated the same conditions for the 12 speaker SI model. Simply pooling the 12 speakers without smoothing does not perform as well as the 109 speaker model. And once again, smoothing the jointly trained model has a rather weak effect on performance. However, we were surprised that the 12 speaker model should have only 25% more error than the 109 speaker model.

The final two results show the effect of independently smoothing the 12 speaker model after either joint or independent training. To independently smooth the jointly trained model, we first trained on the pooled data as usual. Then a SD model was made, for each training speaker, by running the forward-backward algorithm on the combined SI model but on data from only one speaker in turn. This allowed us to generate a set of SD models for

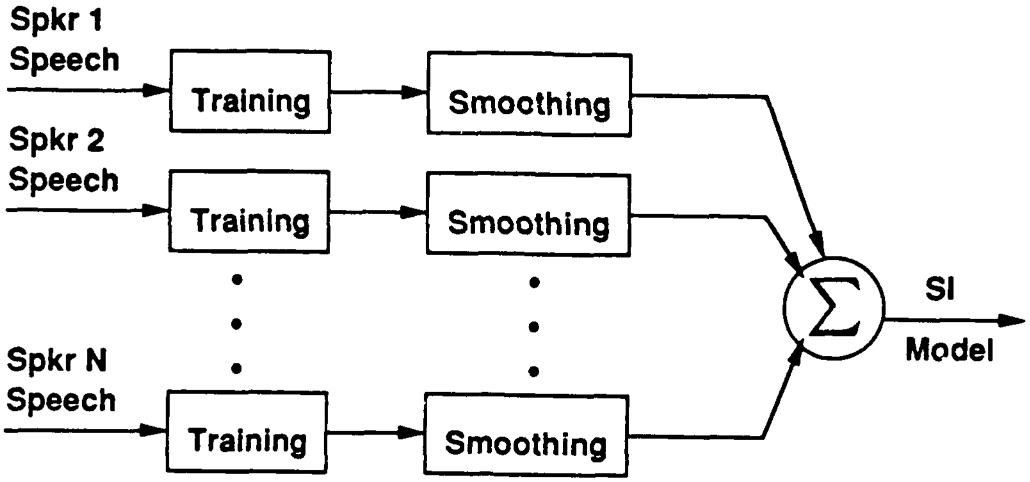


Figure 3.1: Block diagram of independent training and smoothing method.

smoothing, which shared a common alignment. The smoothed models were then recombined by averaging the model statistics.

The approach used on the final result, which is illustrated in Fig. 3.1, is the most straightforward. First, we train a separate SD model from the speech of each speaker independently, allowing each to align optimally for the specific speaker. Second, we smooth each of these models independently to model random spectral variation within the speaker. (Note that these operations can be performed at different times for different speakers as we get more data from new speakers. In addition they can be performed in parallel on different machines if speed is required.) Finally, we combine the models from all the speakers simply by averaging the corresponding discrete probabilities across the speakers.

As is evident from Table 3.1, both of the final methods improve performance, due to the increased effectiveness of the smoothing when it is applied to a speaker-specific model. In a final surprise, we find that constraining all the speakers to a common alignment does not help. Further, the word error rate of this simple model is only 15% worse than the comparable condition with the 109 speaker model and the sentence error rates are statistically indistinguishable.

Some caution is required in comparing results of the 12 and 109 speaker models due to two, possibly important differences. The total amount of training speech used is different as is the number of different sentence texts contained in the training script. The 109 speaker model is trained on a total of 4360 utterances drawn from 2800 sentence texts. The 12 speaker model is trained on 7200 utterances drawn from only 600 sentence texts. While the additional speech may benefit the 12 speaker condition, the greater richness of the sentence texts may help the 109 speaker model. The effect of the additional sentence texts can be seen in the different numbers of triphone contexts observed in the two training scripts: 5000 triphones for 600 sentences vs. 7000 for the 2800–sentence script.

Discussion of SI Results

We have observed that the forward-backward algorithm freely redefines some of the phonemes to model peculiarities of a given speaker. If we constrain all speakers to a common alignment, the training procedure must make a compromise between these speaker-specific adjustments. Both forward-backward and triphone coocurrence smoothing are arguably speaker-specific procedures — they work best when the training distributions are generated by a single source. Some compromise must be made for SI recognition, where the training is not homogeneous and the test distribution is, by definition, different than the training. It appears, from these results, that the least damaging compromise may be to delay pooling of the data/models until the last possible stage in the processing.

Such a simple SI paradigm has several attractive attributes. It makes the data collection effort easier. It is trivial to add new training speakers to the SI model; no retraining is required. Therefore the system can easily make use of any speakers who have already committed to giving enough speech to train a high-performance SD model. There is a large payoff for being one of the training speakers in this scenario — highly accurate SD performance. In contrast, there is no benefit for being a training speaker for the 109 speaker model. Finally, by delaying the stage at which the data or model parameters are pooled, new opportunities arise to use speaker-specific modeling approaches such as the multiple-reference adaptation procedure described in Chapter 4.

Chapter 4

Speaker Adaptation

During the previous three-year effort, we developed a technique for speaker adaptation in which we modified the HMM parameters of one (reference) speaker so that they were appropriate for another (target) speaker, using only 2 minutes of speech from the target speaker. The technique uses a probabilistic spectral mapping from one speaker to another. The mapping is implemented using a probabilistic speaker transformation matrix. We also developed a supervised, text-independent method for estimating the matrix using the Forward-Backward algorithm.

At the end of the previous effort, we devised a new, more effective technique for estimating the speaker transformation matrix. The estimation technique assumes that the sentences in the 2 minutes of speech from the target speaker were also spoken by the reference speaker. (Thus the algorithm is text-dependent.) Then it aligns each pair of corresponding utterances using dynamic time warping (DTW) to estimate the transformation matrix. We also made several improvements to the basic speaker adaptation algorithm. At this time, the recognition accuracy that we observe when the speech of only the reference speaker is adapted to the target speaker is equivalent to that for a speaker-independent system that has been trained on over 100 speakers. Thus, the speaker adaptation scenario represents a dramatic savings in the total effort required to acquire acceptable speech recognition performance in a new recognition domain. In addition to the significant economy argument, speaker adaptation can overcome the very high error rate that occurs when a speaker-independent model is tested on speakers with strong accents or in acoustic environments that are different from those in the training. Our work in single-reference speaker adaptation is described in Section 4.1.

In Section 3.2 we described a new paradigm for speaker-independent training in which the speech of only a dozen speakers is used for training. During this effort, we have also developed a way to use this paradigm as a basis for speaker adaptation from a speaker-independent training set. In this case, by performing the speaker adaptation we can reduce the error rate by 45% relative to the speaker-independent starting point. Our new methods for multiple-reference speaker adaptation are described in Section 4.2.

4.1 Single-Reference Speaker Adaptation

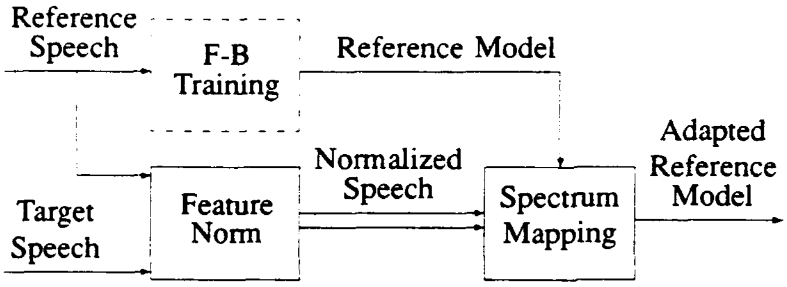


Figure 4.1: Baseline speaker-adaptation system.

Our baseline speaker-adaptation system consists of two distinct components, shown in solid-line boxes in Figure 4.1. Both of these estimate a non-parameteric transformation between the reference and target speaker, with the goal of making one of them 'look' like the other.

The *feature normalization* component estimates a deterministic transformation which is applied to the speech features of the reference speaker. DTW is used to derive an alignment between the spectral feature vectors of a given pair of utterences from different speakers. Corresponding subpopulations of feature vectors across the two speakers are defined by the alignment and by a vector-quantizer (VQ) labeling of the reference speaker's speech. The means of the corresponding subpopulations are then made identical by shifting the feature vectors of the target speaker. This transformation can be applied iteratively since each application of DTW and feature translation reduces (or leaves unchanged) the mean square error of the alignment. After this transformation, the speech features of the reference speaker are superimposed upon the feature space of the target speaker. The normalization procedure is described in detail in [10].

The *spectrum mapping* component estimates a probabilistic transformation which is applied to the reference HMM observation densities. DTW is again used to define a pair-wise correspondence between the VQ spectra of the reference and target speakers' speech. VQ co-occurrence probabilities are estimated from frequency counts of the co-occurring VQ pairs in the alignment and are accumulated into a transformation matrix. The reference HMM (discrete) observation densities are then multiplied by this matrix. After this transformation, the adapted reference model can be used as an approximation to a well-trained HMM for the target speaker. The spectrum mapping procedure is described in [9].

The reference model is created as usual by using the 'forward-backward' algorithm to train context-dependent phonetic models from the SD data of the reference speaker. This is indicated by the dashed-line box in Figure 4.1.

The basic technique for speaker adaptation has remained fixed over most of this contract. That is, we still use the probabilistic spectral mapping method. We have tried several other methods, but have, as yet, not found any to work as well. However, we have made several improvements in the performance of our adaptation method. These improvements derived from using additional speech parameters, from more accurate estimation techniques, and from an improved distance metric for aligning the adaptation and reference speech.

One of the fundamental aspects of the algorithm involves aligning the speech parameters of two different speakers. The speech parameters are quite different to start. Therefore, it is possible that the alignment found using a simple distance measure is not correct. That is, it does not actually align corresponding phonetic events in the two sentences. We developed an iterative alignment procedure to alleviate this problem. The algorithm is described in [10]. The first alignment serves to define a nonparametric deterministic parameter normalization between the two speakers. This normalization is used to make the parameters of the two speakers more similar. Then we perform a second iteration of alignment. (We showed in the paper that the algorithm is guaranteed to converge.) After we have determined the correct alignment, we use the usual technique for estimating the speaker transformation matrix. We reported a reduction in the recognition error rate resulting from the improved alignment.

When we changed the system to use the derivatives of the cepstral parameters in addition to the steady state parameters, we observed the expected improvement in performance that had been observed for speaker-dependent and speaker-independent recognition. However, we also observed that if we used the derivatives in the distance measure used for aligning the pairs of utterances, the alignment between corresponding sentences was significantly improved, resulting in further improvement in recognition accuracy.

The two improvements above led us to believe that the distance measure used during alignment should be considered more carefully. In addition, we needed a way to normalize the measure in order that we could use both steady-state cepstra and their derivatives together in the same measure, even though the derivatives have a much smaller dynamic range. First we tried normalizing each parameter so that it had unit variance. This allowed us to combine the two different kinds of parameters. However, we reasoned that the lower numbered cepstra and their derivatives which specify the overall spectral shape should carry more weight than the higher parameters which specify the fine detail in the spectrum. After examining several proposed metrics, we normalized each cepstrum parameter such that its variance was inversely proportional to its parameter number. For example, the normalized variance of the 1st cepstrum was 1, while the normalized variance of the 2nd cepstrum was 1/2, and the normalized variance of the 14th cepstrum was 1/14. We performed the same normalization on the derivative parameters. We found that the resulting weighted distance metric resulted in better alignment and therefore lower error rate.

Finally, we considered the mathematics of the estimation of the transformation matrix. When we considered the corresponding spectra along the best alignment path between the two sentences, we reasoned that we must be careful to take into account how many frames from one sentence were aligned against the other. For example, if the best path aligned two frames from the target sentence against one frame of the reference sentence, then we only added 1/2 to each of the corresponding bins of the transformation matrix. We found that making sure that we obtained the correct maximum likelihood estimate of the matrix resulted in some reduction of the error rate.

As a result of the improvements described above the word error rate using models adapted from a single speaker has been reduced to 5% on the DARPA corpus, depending on the amount of speech available from the reference speaker. This level of accuracy is essentially equivalent to the best speaker-independent accuracy, at a small fraction of the start-up cost.

4.2 Speaker Adaptation Using Multiple Reference Speakers

In recent years several researchers have demonstrated speaker-independent (SI) recognition using essentially the same recognition algorithms used for speaker-dependent recognition, but with a model derived by simply pooling the training speech of over 100 speakers (using more than 4000 utterances) as if it all were produced by one speaker. While the recognition accuracy is not as high as for speaker-dependent models, pooling the training data from several speakers obviously makes the system capable of dealing better with speech from new speakers.

In an effort to improve on SI performance, we have considered several ways of adapting models derived from several speakers to be useful for a new speaker. We describe two methods for adaptation from multiple reference speakers. The first method compacts the *speech parameters* of each of the reference speakers so that they are more similar, and then adapts the target speaker to this compacted group of speakers. The second method adapts the *HMM model* from each of the reference speakers to the target speaker and then averages the adapted models. The second method was found to result in much larger improvement.

4.2.1 Compacted Model

Our basic speaker adaptive (SA) approach can be used with a pool of many reference speakers if we can overcome two obvious problems. Our baseline system estimates a transformation between two speakers based on a detailed correspondence between their short-term spectra. It is not obvious how to generalize the transformation to make the correspondences between the target speaker and a pool of reference speakers. Also, we know that training data pooled from many speakers yields a model that has very broad (less discriminating) distributions compared to those produced by speaker-dependent training. Since our adaptation procedures also smooth the reference model, we expect that a straightforward application of them to a pooled speaker-independent model will fail to yield improvements due to excessive smoothing.

One solution to these problems is to normalize the speech features of the many reference speakers to a single, common space prior to pooling. This approach attempts to make all the reference speech appear to be from the same virtual speaker. Also, by making the feature space of the pooled speech more compact, the normalized speech should yield a model with less broad (more discriminating) distributions.

Since the feature normalization component of our system is designed to superimpose the speech features of one speaker onto another's, it can be used to transform the features of many reference speakers to a single speaker whom we designate the *prototypical reference* speaker. Figure 4.2 illustrates this approach. The feature normalization is used repeatedly to define transformations between each reference speaker and the prototypical reference speaker. The normalized reference speech is then pooled and trained as if it were produced by a single speaker.

The target speaker is also normalized to the common feature space. A spectrum mapping is then estimated between the target and the prototypical speaker and the transformation is applied to the normalized–pooled reference model.

<u>**Experimental Conditions**</u>

We ran an experiment using the SI portion of the DARPA Resource Management database as the pool of reference speakers. 40 utterances from each of 109 training speakers were used as training material for this experiment. 10 to 15 utterances from each speaker were drawn from the same scripts used for the prototypical reference speaker's data. This amounted to 30–45 seconds of adaptation speech for each of the training speakers.

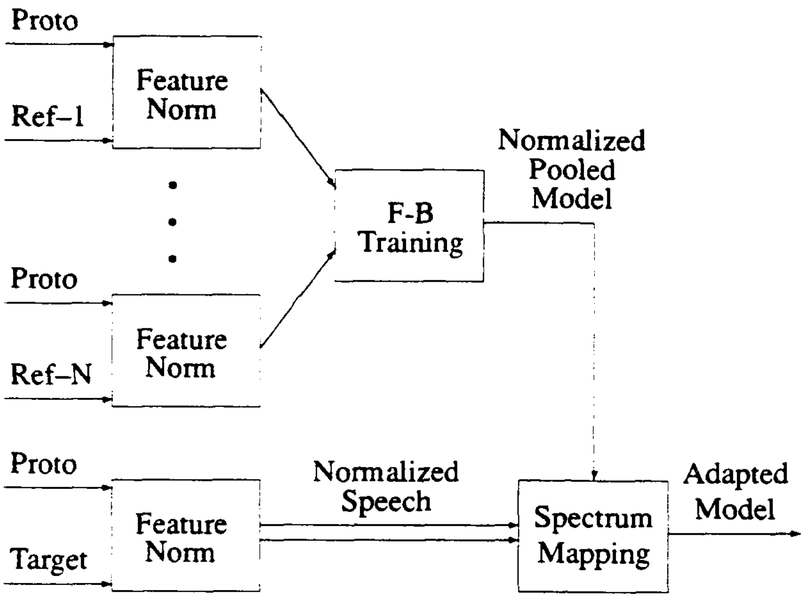


Figure 4.2: Speaker adaptation from a normalized-pooled reference model.

For the prototypical reference speaker, we used speech collected at BBN from speaker RS; the same speaker that is used as the reference in the baseline system. 600 sentences from the SD training script were available from RS as well as the 10 designated 'rapid adaptation' sentences.

The test speakers used are taken from the SD portion of the database so that a direct comparison with our SD results can be made. The database contains 12 such speakers. We did not have adaptation material for speaker ERS at the time of this writing, so this speaker has been omitted from this test, leaving 11 test speakers.

Condition →	Xspkr	Norm	Map	Norm
# Ref Spkrs		Only	Only	+ Map
1 Proto	99	34	8	6.7
109 SI	7.1	8.4		
110 SI + Proto	14.2	6.3		

Table 4.1: Comparison of single and multiple reference speaker-adaptation systems. Numbers shown are perent word error.

In Table 4.1 we show initial results. The standard word-pair grammar (perplexity = 60), defined as part of the database for evaluation purposes, was used in all cases.

The condition labeled, *Xspkr*, signifies that the cross-speaker model (made from one or many speakers not including the target) is used directly and without modification during recognition of the target speaker.

For the condition, *Norm Only*, the cross-speaker model is used for recognition as above, but the test speech of the target speaker is transformed first, using the feature normalization estimated from the adaptation speech. The transformed test data is then quantized using the cross-speaker codebook. This condition allows us to see the effect on performance for the feature normalization component alone.

The third condition, denoted by *Map Only*, indicates that the cross-speaker reference model is transformed by the spectrum mapping procedure before being used in recognition. The test data used is the original unnormalized speech quantized by the target codebook. This condition permits us to see the impact of the spectrum mapping component alone.

The final condition indicates that both transformations are used. For this condition, the feature normalization is used only to improve the alignment for estimating the spectrum mapping. Again, the test data used is the original unnormalized speech.

The table contains results for reference models made from three different training corpora. The label, *1 Proto*, indicates the baseline single reference speaker system of Figure 4.1. The *109 SI* pooled model is made from the 109 training speakers defined by the database. The *110 SI + Proto* model also includes speech from the prototypical reference speaker. The system shown in Figure 4.2 was used to create the two pooled models.

Note that there are two normalization stages in the pooled reference speaker systems: normalization of the many references, and normalization of the target speaker. In Table 4.1 the labeled conditions refer to the target normalization only. For all but the *Xspkr* condition the pooled models are made from normalized reference speech.

Discussion

In the first row of Table 4.1, results for the single reference speaker baseline system are given. The cross-speaker performance is degenerate at approximately 99% word error (1/3 of the words were correctly recognized but many insertion errors were made). Using the feature normalization reduces the error by about a factor of 3. This result is averaged over only 6 of the 11 test speakers used in the rest of the table. Using only the spectrum mapping, however, reduces the cross-speaker error by more than a factor of 10. Despite their unequal power, the two transformations used together appear to improve over their individual results, yielding a 15-fold reduction in error rate over the cross-speaker performance.

It should be noted that the system used for the *Map Only* result did not use the feature pre-conditioning described earlier, so it is possible that the spectrum mapping alone may be adequate for the single reference system.

The second row of the table shows results for the 109 reference speakers from the SI database. The result for simple pooling (*Xspkr*) also improves 15-fold over the single cross-speaker performance. Clearly, pooling the speech of many speakers is powerful but by itself it performs no better than using a speaker transformation on a single reference speaker.

Compacting the 109 speakers and the target speaker by feature normalization has hurt performance somewhat as shown in the *Norm Only* condition. Note that this model does not contain the prototypical reference speech which defined the common space during estimation of the transformations. Still, we would expect a compacted model to do better than a simply pooled one. In fact the overall average probability, given the model, for the compacted data is only slightly higher than for the simply pooled (SI) data and is considerably less than usually achieved for SD training. This result indicates that the feature normalization procedure destroys some information, and in this case, is not powerful enough to overcome that loss.

The last two conditions for the 109 reference model have not been completed at this time. We do not expect spectrum mapping to improve much, however, with the prototypical reference omitted from the training data.

The last row of the table, labeled *110 SI + Proto*, displays results achieved from a model which contains the prototypical speaker in addition to the 109 SI speakers. For the simply pooled condition, performance is much worse than that of the 109 speaker model. This is not surprising given that speech from the prototypical speaker dominates the training data (1/8 of the data) and that this speech, when used in a single cross-speaker reference model, yields degenerate performance.

For the *Norm Only condition*, the 110 speaker model gives marginally better performance than the best result for the other two training corpora. As in the 109 speaker model, the overall average probability of the data was closer to the pooled SI model than to the SD model. It is possible that the feature normalization procedure is more sensitive to the amount of data used to estimate the feature transformation than we anticipated. Recall that we typically use 2 minutes of adaptation speech for the single reference speaker system but we are only using 30–45 seconds of speech for the pooled data.

We have also attempted to use the same SI corpus of over 100 speakers for speaker adaptation as reported in [18]. In this work, we estimated a deterministic transformation on the speech parameters of each of the training speakers which projected them onto the feature space of a single *prototypical* training speaker. We then trained on all of the transformed speech as if it came from a single speaker. The target speaker was similarly projected onto the prototypical speaker and recognition proceeded using the prototypical model. This procedure reduced the word error rate by 10% compared to the SI result; a minor improvement for a significant increase in the complexity of the scenario. We believe that this method did no better because the feature transformation was not powerful enough to superimpose a pair of speakers without significant loss of information. This resulted in a prototypical model whose densities were not significantly sharper than the comparable SI model made from the original data.

4.2.2 Adaptation from 12 Speakers

Our experience with the 109 corpus led us to rethink our approach to speaker adaptation from multiple reference speakers.

We already have a powerful speaker adaptation procedure which effectively transforms a single well-trained SD reference model into an adapted model of the target speaker [9]. The transformation is estimated from a small amount of adaptation data (40 utterances) given by the target speaker. The approach is powerful for two reasons: first, the estimate of the probabilistic spectral mapping between two speakers is robust and generalizes well to phonetic contexts not observed in the adaptation speech, and second, the transformation can be applied to the well-estimated, discriminating densities of the SD reference model without undue loss of detail.

A natural extension of this approach to multiple references would be to combine the parameters of several SD models after they had been independently adapted to the same target speaker. We can assume from our 12 speaker SI experiments that the transformation will perform better if estimated independently between each speaker-pair in turn rather than from a pooled dataset, since the transformation is a speaker-pair-specific operation. We also know that we can successfully combine the multiple adapted models by averaging the model statistics.

4.2.3 Experimental Results

Test Conditions

In these experiments, we have used the 12 RM1 speakers as test speakers (as well as references in some experiments). The test data consists of 25 different utterances for each speaker. The entire test set is composed of 300 utterances and contains more than 2400 word tokens. In all cases, the first 40 sentences of the common training material (about 2 minutes of speech) is used as adaptation data.

Results

To establish a baseline, we used each of the four RM2 speakers with 30 minutes of training as single references. In Table 4.2, we show an average performance of 6.5% word error rate for all combinations of the 4 references and the 12 test speakers. The sample standard deviation of the individual target speaker results is 4.3%, indicating the wide variation in performance over the test speakers, which ranged from 0.6% to 18.7%.

Condition	% Word Err	Std. Dev.
Single reference baseline	6.5	4.3
11 reference, uniform wgts	4.6	3.0
11 reference, estimated wgts	4.1	2.5

Table 4.2: Comparison of adaptation performance from single and multiple r^ferences.

Two additional results are shown in Table 4.2 for averaging the adapted models of 11 reference speakers. For these experiments, we jackknifed over the 12 RM1 speakers holding out one test speaker at a time. Each of the 11 reference models was trained using 30 minutes of speech. The multi-reference model made with uniform weights shows a substantial reduction in both word error rate and variablility compared to the baseline. The model using weights estimated from the adaptation data improved further, resulting in a 37% overall reduction in error rate compared to the baseline. Most of the improvement gained by weighting the adapted reference models is due to a substantial improvement in the worst speaker.

In another set of experiments, we investigated the effect of using more than 30 minutes of speech for training each reference model. As shown in Table 4.3, repeating the single reference baseline with 2 hours of speech for each reference speaker yields a 10% reduction in word error averaged over all combinations of reference and target. When the 4 RM2 reference models are used in weighted combination, the error is further reduced by 25%, equalling the error rate of the 11 reference result given above. Note, however, that the 4 reference model has not corrected the problem with the outlier speaker as indicated by the deviation in individual speaker performance.

Using 2 hours of data instead of 30 minutes to train the reference models did improve

Condition	% Word Err	Std. Dev.
Single reference baseline	5.8	3.3
4 reference, estimated wgts	4.2	3.7

Table 4.3: Adaptation performance with references trained from 2 hours of data.

the average results for every reference speaker. However, only 60% of the target speakers improved while 25% actually degraded. This led to a small improvement overall — only 12% error reduction for a quadrupling of the reference training data. We have concluded that 30 minutes of training data from each reference speaker is adequate for this task.

In Table 4.4, we show the RM2 single reference baseline results again, but separated into the four possible combinations of gender pairs. It is notable that the gender of the reference speaker has little effect upon performance, and no effect on the female test speakers. For this test set, the male speakers tend to be somewhat worse speakers on average.

	Genders of Speaker Pair	
Gender of Target	Same	Opposite
Female	4.0	4.2
Male	6.5	7.5

Table 4.4: Effect of reference gender on single reference performance.

Discussion

To get a sense of how well our basic adaptation procedure models the difference between two speakers, we can compare the single reference adaptation results to cross-speaker recognition (train on one speaker, test on another, without adaptation). Measured on the same test set as was used in Table 4.2, the cross-speaker error rate is quite bad — about 77%. By comparison, our baseline single-reference performance is 12-fold better.

We discovered that this result is misleading, however, since the effect of cross-speaker gender is very strong in this case. Repeating the experiment with two cross-speaker models, one from each gender, we found that matching genders between training and test speakers reduces the error rate to 15%. Furthermore, we found that the correct gender model could be chosen automatically, with no degradation in performance, by comparing the probability of an utterance as measured by each of the two models. So our baseline adaptation procedure, which is implicitly gender-independent, improves over gender-dependent cross-speaker performance by better than a factor of two.

We have shown that we can increase this improvement over cross-speaker recognition to nearly a factor of four by using training speech from multiple references. It is natural then to compare this performance with an analogous cross-speaker model made from multiple training (reference) speakers.

4.3 Conclusion

We have shown that our baseline speaker adaptation algorithm gives consistent performance, on a 1000-word continuous speech recognition task, across a wide variety of reference speakers including speakers of opposite gender from the target. Needing only 600 utterances to train the reference speaker, this approach provides an economical and logistically simple way to bring up new users quickly to a high level of performance on an arbitrary task domain.

In addition, we have demonstrated a speaker adaptation procedure using multiple reference speakers which performs much better than adaptation from a single reference. The nearly 40% reduction in error rate due to the use of multiple references is larger than previously reported in the literature on speaker adaptation.

Bibliography

[1] S. Austin, C. Barry, Y-L. Chow, A. Derr, O. Kimball, F. Kubala, J. Makhoul, P. Placeway, W. Russell, R. Schwartz, and G. Yu (1989) "Improved HMM Models for High Performance Speech Recognition," *Proceedings of the DARPA Speech and Natural Language Workshop*, October, 1989.

[2] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer (1986) "Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition." *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, 1986, Vol 1, pp. 49-52, Paper No. 2.3.1.

[3] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer (1988) "A New Algorithm for the Estimation of Hidden Markov Model Parameters", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, New York, April, 1988, pp. 493-496, Paper 11.2.

[4] L.Baum (1972) "An Inequality and association Maximization technique in Statistical Estimation for Probabilistic Function of Markov Processes," *Inequality*, Vol III, 1972, pp 1-8.

[5] Bellagard, J. and D. Nahamoo (1989) "Tied mixture continuous parameter models for large vocabulary isolated speech recognition", *IEEE ICASSP89*.

[6] Brown, P. (1987) "The Acoustic-Modeling Problem in Automatic Speech Recognition", *PhD Thesis*, CMU, 1987.

[7] Chow, Y-L. and Schwartz, R.M. (1990) "The N-Best Algorithm: An Efficient Procedure for Finding Top N Sentence Hypotheses", *Proceedings of the DARPA Speech and Natural Language Workshop*, Cape Cod, October 1989.

[8] G. Doddington (1989) "Phonetically Sensitive Discriminants for Improved Speech Recognition," *IEEE ICASSP-89*, pp. 556-559.

[9] Feng, M., F. Kubala, R. Schwartz, J. Makhoul (1988) "Improved Speaker Adaptation Using Text Dependent Spectral Mappings", *IEEE ICASSP-88*, Apr. 1989, paper S3.9.

[10] Feng, M., R. Schwartz, F. Kubala, J Makhoul (1989) "Iterative Normalization for Speaker-Adaptive Training in Continuous Speech Recognition," *IEEE ICASSP-89*, paper S12.4.

[11] Furui, S. (1986) "Speaker-Independent Isolated Word Recognition Based on Emphasized Spectral Dynamics," *IEEE ICASSP-86*, pp. 1991-1994.

[12] Furui, S. (1989) "Unsupervised Speaker Adaptation Method Based on Hierarchical Spectral Clustering", *IEEE ICASSP-89*, May 1989, paper S6.9.

[13] Gupta, V.N., Lennig, M., Mermelstein, P. (1987) "Integration of Acoustic Information in a Large Vocabulary Word Recognizer", *IEEE ICASSP-87*, Apr. 1987, pages 697-700.

[14] Hattori, H., S. Nakamura, K. Shikano (1990) "Supplementation of HMM for Articulatory Variation in Speaker Adaptation", *IEEE ICASSP-90*, Apr. 1990, paper S3.6.

[15] Huang, X.D. and M.A. Jack (1989) "Semi-continuous hidden Markov models for speech recognition", *Computer Speech and Language*, Vol 3, 1989.

[16] Jelinek, F., Mercer, R.L. (1980) "Interpolated Estimation of Markov Source Parameters from Sparse Data" in E.S. Gelsema and L.N. Kanal (editor), *Pattern Recognition in Practice*, pages 381-397, North-Holland Publishing Company, Amsterdam, 1980.

[17] Kohonen, T., G. Barna, and R. Chrisley (1988) "Statistical Pattern Recognition with Nerual Networks: Benchmarking Studies," *IEEE Proc. of ICNN*, Vol. I, pp. 61-68, July 1988.

[18] Kubala, F., R. Schwartz, C. Barry, "Speaker Adaptation from a Speaker-Independent Training Corpus", *IEEE ICASSP-90*, Apr. 1990, paper S3.3.

[19] Lee, K. (1988) "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System", PhD dissertation, Carnegie-Mellon University, Apr. 1988, CMU-CS-88-148.

[20] Lee, K., H. Hon, M. Hwang (1989) "Recent Progress in the Sphinx Speech Recognition System", *Proceedings of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann Publishers, Inc., Feb. 1989, pp. 125–130.

[21] Makhoul, J., S. Roucos, and H. Gish (1985) "Vector Quantization in Speech Coding", *Proc. IEEE*, Vol. 73, No.11, pp.1551-1588.

[22] McDermott, E. and S. Katagiri (1989) "Shift-Invariant, Multi-Category Phoneme Recognition using Kohonen's LVQ2," *IEEE ICASSP-89*, pp. 81-84.

[23] A. Nadas, "A decision-theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional verus conditional maximum likelihood," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Volume ASSP-31, Number 4, pages 814-817, August 1983.

[24] Paul, D.B., personal communication, Feb. 1988.

[25] Nakamura, S. and K. Shikano (1989) "Speaker Adaptation Applied to HMM and Neural Networks", *IEEE ICASSP-89*, May, 1989, paper S3.3.

[26] Price, P., Fisher, W.M., Bernstein, J., and D.S. Pallett (1988) "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, April 1988, pp. 651-654.

[27] Rigoll, G. (1989) "Speaker Adaptation for Large Vocabulary Speech Recognition Systems Using Speaker Markov Models", *IEEE ICASSP89*, May, 1989, paper S1.2, pp. 5-8.

[28] Rtischev, D. (1989) "Speaker Adaptation in a Large-Vocabulary Speech Recognition System", Masters thesis, Massachusetts Institute of Technology, Jan. 1989.

[29] Schwartz, R.M., Chow, O., Roucos, S., Krasner, M., and J. Makhoul (1984) "Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition", *Proceedings ICASSP 84*, paper 35.6, March, 1984.

[30] Schwartz, R.M., Chow, Y., Kimball, O., Roucos, S., Krasner, M., and J. Makhoul (1985) "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech", *Proceedings ICASSP 85*, pp. 1205-1208, March, 1985.

[31] Y.L. Chow, R. M. Schwartz, S. Roucos, O.A. Kimball, P.J. Price, G.F. Kubala, M.O. Dunham, M.A. Krasner, and J. Makhoul (1986) "The Role of Word-Dependent Coarticulatory Effects in a Phoneme-Based Speech Recognition System", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, April 1986, pp. 1593-1596, Paper No. 30.9.

[32] Schwartz, R., Y. Chow, F. Kubala (1987) "Rapid Speaker Adaptation using a Probabilistic Spectral Mapping", *IEEE ICASSP-87*, Apr. 1987, paper 15.3.1.

[33] R. Schwartz, Y-L. Chow, A. Derr, M-W. Feng, O. Kimball, F. Kubala, J. Makhoul, M. Ostendorf, P. Price, and S. Roucos (1988) "Statistical Modeling for Continuous Speech Recognition," BBN Report No. 6725, Bolt Beranek & Newman Inc., Cambridge, MA, February 1988.

[34] Schwartz, R., O. Kimball, F. Kubala, M. Feng, Y. Chow, C. Barry, J. Makhoul (1989) "Robust Smoothing Methods for Discrete Hidden Markov Models", *IEEE ICASSP-89*, May 1989, paper S10b.9.

[35] Schwartz, R. and Y.L. Chow (1990) "The N-Best Algorithm: An Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses", *ICASSP-90*, April 1990, Albuquerque S2.12, pp. 81-84.

[36] K. Shikano, K.F. Lee, and R. Reddy (1986) "Speaker Adaptation Through Vector Quantization", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, April 1986, pp. 2643-2646, Paper No. 49.5.

[37] K. Sugawara, M. Nishimura, K. Toshioka, M. Okochi, and T. Kaneko (1985) "Isolated Word Recognition Using Hidden Markov Models", *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa FL, March 1985, pp. 1-4.

[38] Tseng, P., Sabin, M., and E. Lee (1987) "Fuzzy Vector Quantization Applied to Hidden Markov Modeling", *IEEE ICASSP-87*, April, 1987, paper S15.5.